

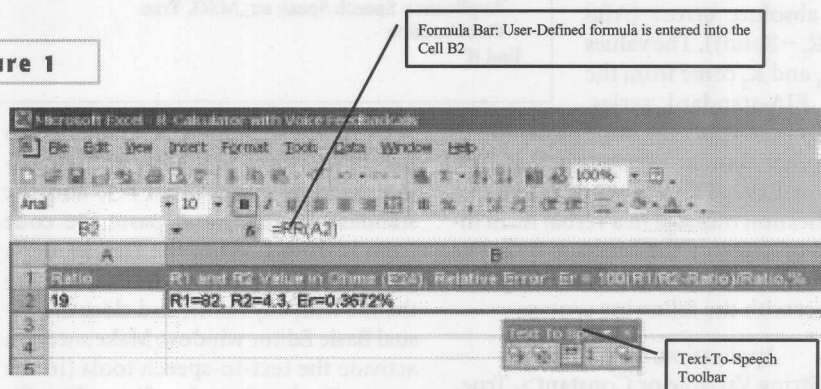
# Voice feedback enhances engineering calculator

Alexander Bell, InfoSoft International Inc, Rego Park, NY

**T**HE SCREEN SHOT IN Figure 1 represents a Microsoft Excel 2002 worksheet designed to im-

plement VFI (voice-feedback interface) for an engineering calculator. The voice-interface technique has both practical and educational aspects. It automates the common task of finding the values of two resistors for a given ratio. It also demonstrates the latest advances in natural-language programming technology with an example of the technology's actual implementation in CAD/CAE systems. A single user-defined function RR (resistor ratio) encapsulates both the computation engine and the VFI. The function uses VBA (Visual Basic for Applications) with the code placed in the standard code module of Excel File (Listing 1). You can download Listing 1 and the Excel work-

Figure 1



Cell A2 serves for data (ratio) entry; cell B2 contains the user-defined formula 5RR(A2). In automatic calculation mode, every time you enter a new ratio value in A2, the system recalculates R<sub>1</sub>, R<sub>2</sub>, and Relative Error.

sheet from the Web version of this Design Idea at [www.ednmag.com](http://www.ednmag.com).

The core search algorithm, which contains outer and inner loops, sequentially

## LISTING 1—CODE MODULE FOR VOICE FEEDBACK

### Option Explicit

Function RR(ByVal Ratio As Variant) As String

\*\*\* AUTHOR: ALEXANDER BELL

\*\*\* DATE : 12/02/02

\*\*\* USAGE : FIND THE RESISTORS' VALUE R1 AND R2 FROM E24 SERIES WHICH MAKES THE BEST

\*\*\* : APPROXIMATION FOR GIVEN RATIO, BASED ON CRITERIA: MIN(ABS(RATIO-R1/R2)).

\*\*\* : FORMULA LIMITATIONS: 1<= RATIO < 100 000 000.

\*\*\* : NOTE: RETURN RESISTORS' VALUES ARE SHOWN IN OHMS

On Error GoTo ErrHandler

Const MAX\_ORDER As Integer = 8 \*\*\* CONST USED TO SET LIMIT FOR RATIO (<100 000 000)

Dim R1 As Double, R2 As Double \*\*\* RESISTORS' VALUE VARIABLES

Dim dbError As Double \*\*\* ERROR OF APPROXIMATION VARIABLE

Dim I As Integer, J As Integer \*\*\* LOOP COUNTERS

Dim dbMantissa As Double \*\*\* MANTISSA VARIABLE

Dim mOrder As Integer \*\*\* ORDER VARIABLE

Dim arrE24 As Variant \*\*\* VARIABLE REFERENCING TO E24 VALUES ARRAY

Dim str\_Msg As String \*\*\* VARIABLE CONTAINING THE VOICE NOTIFICATION TO USER

\*\*\* DATA VALIDATION WITH VOICE FEEDBACK

\*\*\*\*\*

If CSng(Ratio) = "" Then

str\_Msg = "PLEASE, ENTER THE NUMERIC VALUE"

Application.Speech.Speak str\_Msg, True

Exit Function

End If

If Not IsNumeric(Ratio) Then

str\_Msg = "WRONG ENTRY: PLEASE, ENTER THE NUMERIC VALUE"

Application.Speech.Speak str\_Msg, True

Exit Function

End If

If Ratio < 1 Then

str\_Msg = "WRONG ENTRY: RATIO MUST BE MORE OR EQUAL 1, OTHERWISE USE THE INVERSE VALUE"

Application.Speech.Speak str\_Msg, True

Exit Function

End If

If Ratio >= 10 ^ MAX\_ORDER Then

str\_Msg = "WRONG ENTRY: RATIO MUST BE LESS THAN " & CSng(10 ^ MAX\_ORDER)

Application.Speech.Speak str\_Msg, True

Exit Function

End If

\*\*\* E24 SERIES ARRAY. LAST VALUE (10) IS INCLUDED TO SIMPLIFY THE SEARCH

### ALGORITHM

arrE24 = Array(1, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.8, 2, 2.2, 2.4, 2.7, ...  
3, 3.3, 3.9, 4.3, 4.7, 5.1, 5.6, 6.2, 6.8, 7.5, 8.2, 9.1, 10)

\*\*\* CALCULATE ORDER AND MANTISSA FOR GIVEN RATIO

mOrder = Int(Log(Ratio) / Log(10))

dbMantissa = Ratio / (10 ^ mOrder)

\*\*\* SET RESISTORS INITIAL VALUES

R1 = 1: R2 = 1

dbError = Abs(R1 / R2 - dbMantissa)

\*\*\* LOOP THROUGH E24 ARRAY TO FIND R2 AND R1 MANTISSA RESULTING IN A MIN ABS ERROR

\*\*\* NOTE: THIS ALGORITHM RETURNS R1 MANTISSA WHICH IS ALWAYS BIGGER OR EQUAL R2

For I = LBound(arrE24) To UBound(arrE24)

For J = LBound(arrE24) To I

If Abs(arrE24(I) / arrE24(J) - dbMantissa) < dbError Then

R1 = arrE24(I)

R2 = arrE24(J)

dbError = Abs(arrE24(I) / arrE24(J) - dbMantissa)

End If

Next J

Next I

\*\*\* CALCULATE R1 VALUE IN OHMS AND RELATIVE ERROR IN (%)

R1 = R1 \* 10 ^ mOrder

dbError = (R1 / R2 - Ratio) / Ratio

\*\*\* RETURN RESULT STRING: RESISTORS' VALUES IN OHMS, RELATIVE ERROR IN %

RR = "R1=" & CSng(R1)

& ", R2=" & CSng(R2)

& ", Er=" & Format(dbError, "0.0000%")

\*\*\* SEND "OK" VOICE NOTIFICATION TO USER AND EXIT FUNCTION

str\_Msg = "OK"

Application.Speech.Speak str\_Msg, True

Exit Function

ErrHandler:

\*\*\* IN CASE OF ERROR RETURN EMPTY STRING AND SEND THE ERROR VOICE NOTIFICATION TO USER

RR = ""

str\_Msg = "UNANTICIPATED ERROR IN CALCULATIONS"

Application.Speech.Speak str\_Msg, True

End Function

tests each pair of values,  $R_1$  and  $R_2$ , to find the best approximation of the target ratio. In other words, the algorithm tries to minimize the absolute error:  $(ABS(R_1/R_2 - Ratio))$ . The values of  $R_1$  and  $R_2$  come from the E24 EIA-standard series, but you can apply the same algorithm to any other standard series, such as E48, E96, or E192. The VFI sends the status-notification message in a verbal form instead of showing the Message Box. The VFI uses the built-in Excel 2002 Speech Object with the following syntax:

```
Application.Speech.Speak  
<String Variable or Constant>, True,
```

where <String Variable or Constant> contains the actual spoken text, and the second property is set to True for asynchronous mode. **Listing 2** is an example of voice-error notification in the case that you enter non-numeric data as the ratio (data-validation error message):

## LISTING 2—VOICE-ERROR NOTIFICATION

```
IfNot IsNumeric(Ratio) Then  
Str_Msg5"Wrong ENTRY: PLEASE, ENTER THE NUMERIC VALUE."  
Application.Speech.Speak str_Msg, True  
Exit Function  
End If
```

Using the technique is simple. Open Excel File, switch to the Visual Basic Editor window (press Alt-F11), add the standard module, and paste the code from **Listing 1**. From the Debug menu item, choose Compile VBA Project, save the file with any name, and close the Visual Basic Editor window. Make sure you activate the text-to-speech tools (in the menu: Tools—Speech—Show Text To Speech Toolbar). Check whether you are in automatic- or manual-calculation mode (in the menu: Tools—Options—Calculation). Set the mode to automatic; otherwise, you'd have to use the F9 key to force a new calculation every time you enter a new ratio value. Choose any cell

(for example, A2 in **Figure 1**) for the ratio (data-entry cell) and another cell (for example, B2) to display the results. Enter the formula  $=RR(A2)$  into cell B2. Now, every time you enter a new ratio value into

cell A2, the system automatically calculates and displays  $R_1$ ,  $R_2$ , and Relative Error, and sends the status voice notification. The notification is either "OK" to confirm the successful completion of the calculation or an error notification in the case of a data-validation or computation error. Note that some macros in Microsoft Office applications could result in potentially dangerous and harmful actions, and some may contain viruses. You use the macros at your own risk without warranties of any kind.

**Is this the best Design Idea in this issue?** Select at [www.ednmag.com](http://www.ednmag.com).